MARTINI WG
Internet Draft
Intended status: Standards-Track
Expires: April 25, 2011

H. Kaplan
Acme Packet

October 25, 2010

SIP Verification with Event-package for Resolution of Managed Open-ended Username Target Handles (VERMOUTH)

draft-kaplan-martini-vermouth-01

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 25, 2011.

Copyright and License Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

Abstract

The Martini Working Group is defining a mechanism for SIP IP-PBX type devices to REGISTER and obtain SIP service for E.164-based Address of Records, using the GIN mechanism defined in [draft-gin]. Two other drafts, [draft-olive] and [draft-glass], propose the same for non-E.164-based AoRs. This document defines a means by which the IP-PBX can verify the resolution entries in the SSP for open-ended or full AoRs of any GIN-based mechanism, using a new Event-Package named "vermouth".

Table of Contents

1. Introduction

In many deployed SIP Service Provider (SSP) architectures, it is common to use REGISTER requests to provide the reachability information for IP-PBXs, instead of DNS-based resolution and routing. An IETF-defined mechanism for doing so is defined in [draft-gin]. Another draft, [draft-olive], uses the [draft-gin] GIN mechanism for Local-Number AoRs as well; and a new draft [draft-glass] does the same for literal alpha-numeric/email-style AoRs.

In all cases, the IP-PBX or another SIP entity may wish to learn about all of the AoRs which were implicitly Registered by [draft-gin] or [draft-olive], or to learn about changes in their provisioned AoRs through asynchronous notifications. Even in non-Registration scenarios, where requests for specific AoRs in a SSP may instead be statically routed to an IP-PBX, it may be useful for the IP-PBX to learn what those AoRs are in order to detect mismatches or changes.

In theory, the [draft-gin] mechanism is simply a short-hand single REGISTER transaction for a bulk set of AoRs in lieu of multiple, separate REGISTER transactions for each AoR. In practice, however, the E.164 user numbers may be an "open" numbering plan/range, such that the SSP only really knows about a certain number of digits and the rest are only known to the IP-PBX. Likewise, when [draft-olive] is used, the Local-Number may be only partially known to the SSP.

Therefore, it is not possible for the SSP to actually provide state information for each possible unique AoR instance. Instead, it needs to provide an indication for the registration state of the prefix or digit portion it does know about.

This document proposes to provide such information using a new Event-Package.

2. Definitions

For brevity’s sake, this document uses the word "request" instead of "out-of-dialog request", but in all case means out-of-dialog request.

AoR: address-of-record, as defined by RFC 3261: a URI by which the user is canonically known (e.g., on their business cards, in the From header field of their requests, in the To header field of REGISTER requests, etc.).

Bulk-AoR: a SIP or SIPS address-of-record with a "range" URI user parameter which expands the user string based on a heuristic.

Local-Number: an AoR which follows the form of local-number in [RFC3966], but may be encoded in a SIP or TEL URI. The local-number contains a ’phone-context’ parameter identifying the scope of its number.

Email-style URI: a SIP AoR which does not identify a global E.164 number or Local-Number.

Implicit Registration: implicitly providing the reachability information for something other than the AoR explicitly indicated in the Register transaction.

Reachability Information: a set of URI’s identifying the host and path of Proxies to reach that host; like any URI, these URI’s may identify the specific connection transport, IP Address, and port information, or they may only identify FQDN’s.

SSP: SIP Service Provider, as defined by [RFC5486].

3. The Solution - an Overview

The general concept is a SIP device, such as an IP-PBX, Subscribes to a new "vermouth" Event-Package by issuing a SUBSCRIBE request targeted at the SIP URI AoR it explicitly registered using GIN, or some other mutually-agreed-upon SIP-URI if GIN was not used.

If the Subscription is successful, the returned NOTIFY contains a userinfo XML document that lists all of the usernames of the AoR’s domain that the SSP will route to the IP-PBX. The XML document does not contain the Contact/Path routing reachability information, since that information is already in the reg-event package information for the explicitly registered AoR of the IP-PBX, and may also be more sensitive in nature.

To handle the open-numbering-plan problem, an XML "range" attribute is used, which is similar to a regular expression pattern but with a very limited, specified syntax. The limited syntax is used to avoid ambiguities and reduce confusion - rationale for this is provided in Appendix A.

Furthermore, this document specifies that the To-URI used for the [draft-gin] REGISTER request, be usable as the target for the SUBSCRIBE request, both for the new ’vermouth’ Event-Package, and for Subscribing to the [RFC3680] registration event-package for that explicitly registered AoR.

4. Event Package Definition

This section fills in the details needed to specify an event package as defined in Section 4.4 of [RFC3265].

4.1. Event Package Name

The SIP Events specification requires package definitions to specify the name of their package or template-package.

The name of this package is "vermouth". As specified in [RFC3265], this value appears in the Event header present in SUBSCRIBE and NOTIFY requests.

4.2. Event Package Parameters

The SIP Events specification requires package and template-package definitions to specify any package specific parameters of the Event header that are used by it.

No package specific Event header parameters are defined for this event package.

4.3. SUBSCRIBE Bodies

The SIP Events specification requires package or template-package definitions to define the usage, if any, of bodies in SUBSCRIBE requests.

A SUBSCRIBE for registration events MAY contain a body. This body would serve the purpose of filtering the subscription. The definition of such a body is outside the scope of this specification.

A SUBSCRIBE for the registration package MAY be sent without a body. This implies that the default registration filtering policy has been requested. The default policy is:

- o Notifications are generated every time there is any change in the state of any of the registered contacts for the resource being subscribed to. Those notifications only contain information on the contacts whose state has changed.

- o Notifications triggered from a SUBSCRIBE contain full state (the list of all contacts bound to the address-of-record).

Of course, the server can apply any policy it likes to the subscription.

4.4. Subscription Duration

The SIP Events specification requires package definitions to define a default value for subscription durations, and to discuss reasonable choices for durations when they are explicitly specified.

The Event Package defined herein is not tied to registration state, nor to any value that has natural expiry times. Therefore, the suggested subscription duration is 86400 seconds (1 day).

Of course, clients MAY include an Expires header in the SUBSCRIBE request asking for a different duration.

4.5. NOTIFY Bodies

The SIP Events specification requires package definitions to describe the allowed set of body types in NOTIFY requests, and to specify the default value to be used when there is no Accept header in the SUBSCRIBE request.

The body of a notification of a change in provisioned usernames contains a user information document. This document describes some or all of the username expansions associated with the particular address-of-record subscribed to. All subscribers and notifiers MUST support the "application/userinfo+xml" format described in Section 5. The subscribe request MAY contain an Accept header field. If no such header field is present, it has a default value of "application/userinfo+xml". If the header field is present, it MUST include "application/userinfo+xml", and MAY include any other types capable of representing registration information.

Of course, the notifications generated by the server MUST be in one of the formats specified in the Accept header field in the SUBSCRIBE request.

4.6. Notifier Processing of SUBSCRIBE Requests

The SIP Events framework specifies that packages should define any package-specific processing of SUBSCRIBE requests at a notifier, specifically with regards to authentication and authorization.

Provisioned usernames can be sensitive information. Therefore, all subscriptions to it SHOULD be authenticated and authorized before approval. Authentication MAY be performed using any of the techniques available through SIP, including digest, S/MIME, TLS or other transport specific mechanisms [1]. Authorization policy is at the discretion of the administrator, as always. However, a few recommendations can be made.

It is RECOMMENDED that an IP-PBX be allowed to subscribe to its own provisioned usernames. Such subscriptions are useful for detecting errors and changes.

4.7. Notifier Generation of NOTIFY Requests

The SIP Event framework requests that packages specify the conditions under which notifications are sent for that package, and how such notifications are constructed.

Instead of delivering the full list every time a notification is sent, it is RECOMMENDED that notifications only list the username entries that have changed state (i.e., been added or removed).

Notifications triggered as a result of a fetch operation (a SUBSCRIBE with Expires of 0) or a new Subscription SHOULD result in the full list of all usernames to be present in the NOTIFY.

4.8. Subscriber Processing of NOTIFY Requests

The SIP Events framework expects packages to specify how a subscriber processes NOTIFY requests in any package specific ways, and in particular, how it uses the NOTIFY requests to construct a coherent view of the state of the subscribed resource.

Typically, the NOTIFY will only contain information for usernames whose state has changed. To construct a coherent view of the total state of all usernames, the subscriber will need to combine NOTIFYs received over time. The details of this process depend on the document format used to convey registration state. Section 5 outlines the process for the application/userinfo+xml format.

4.9. Handling of Forked Requests

The SIP Events framework mandates that packages indicate whether or not forked SUBSCRIBE requests can install multiple subscriptions.

Provisioned usernames are normally stored in some repository (whether it be co-located with a proxy/registrar or in a separate database). As such, there is usually a single place where the username information for a particular address-of-record is resident. This implies that a subscription for this information is readily handled by a single element with access to this repository. There is, therefore, no compelling need for a subscription to username information to fork. As a result, a subscriber MUST NOT create multiple dialogs as a result of a single subscription request. The required processing to guarantee that only a Section 4.4.9 of the SIP single dialog is established is described in Events framework [RFC3265].

4.10.   Rate of Notifications

The SIP Events framework mandates that packages define a maximum rate of notifications for their package.

For reasons of congestion control, it is important that the rate of notifications not become excessive. As a result, it is RECOMMENDED that the server not generate notifications for a single subscriber at a rate faster than once every 5 seconds.

4.11. State Agents

The SIP Events framework asks packages to consider the role of state agents in their design.

State agents have no role in the handling of this package.

5. Username Information

5.1. Structure of Username Information

Username information is an XML document [4] that MUST be well-formed and SHOULD be valid. Username information documents MUST be based on XML 1.0 and MUST be encoded using UTF-8. This specification makes use of XML namespaces for identifying registration information documents and document fragments. The namespace URI for elements defined by this specification is a URN [5], using the namespace identifier ietf defined by [6] and extended by [7]. This URN is:

urn:ietf:params:xml:ns:userinfo

A username information document begins with the root element tag "userinfo". It consists of any number of "userlist" sub-elements, each of which contains the provisioning state for a particular list of usernames, associated with the address-of-record subscribed to. The username information for a particular address-of-record MUST be contained within a single "userlist" element; it cannot be spread across multiple "userlist" elements within a document. Other elements from different namespaces MAY be present for the purposes of extensibility; elements or attributes from unknown namespaces MUST be ignored.

There are two attributes associated with the "userinfo" element, both of which MUST be present:

version: This attribute allows the recipient of username information documents to properly order them. Versions start at 0, and increment by one for each new document sent to a subscriber. Versions are scoped within a subscription. Versions MUST be representable using a 32 bit integer.

state: This attribute indicates whether the document contains the full list of provisioned usernames, or whether it contains only information on those registrations which have changed since the previous document (partial).

Note that the document format explicitly allows for conveying information on multiple addresses-of-record. This enables subscriptions to groups of usernames, where such a group is identified by some kind of URI. For example, a domain might define sip:allusers@example.com as a subscribe-able resource that generates notifications when the provisioning state of any address-of-record in the domain changes.

The "userlist" element has a list of any number of "user" sub-elements, each of which contains information on a single username entry, which may itself be a range-patterned name. Other elements from different namespaces MAY be present for the purposes of extensibility; elements or attributes from unknown namespaces MUST be ignored.

There are three attributes associated with the "userlist" element, all of which MUST be present:

aor: The aor attribute contains a URI which is the address-of-record this list is associated with.

id: The id attribute identifies this list. It MUST be unique amongst all other id attributes present in other userlist elements conveyed to the subscriber within the scope of their subscription. Furthermore, the id attribute for a "userlist" element for a particular address-of-record MUST be the same across all notifications sent within the subscription.

state: The state attribute indicates the state of the username list. The valid values are "active" and "removed".

The "user" element contains the username. There are several attributes associated with the "contact" element which MUST be present:

id: The id attribute identifies this user name. It MUST be unique amongst all other id attributes present in other user elements conveyed to the subscriber within the scope of their subscription.

state: The state attribute indicates the state of the user name. The valid values are "active" and "removed".

type: The type attribute identifies the user name type. Valid values are "e614", "private", and "alpha".

range: the range attribute is defined in the next section.

context: the context attribute is only meaningful when the type attribute is "private", and in such a case the context identifies the context of the private name space.

5.2. The "range" Attribute

The range attribute’s value defines the expansion of the username, using a syntax similar to regular expressions. The range pattern applies after the last character of the user element’s value.

```
range-value    = exp-char-set exp-char-count

exp-char-set   = digit-char-set / any-char-set
digit-char-set = "[" dsc-begin "-" dsc-end "]"
dsc-begin      = DIGIT
dsc-end        = DIGIT
any-char-set   = "."

exp-char-count = "{" exp-min "," exp-max "}"
exp-min        = DIGIT
exp-max        = DIGIT
```

The "digit-char-set" defines a range of digit characters, for example 0-9 or 3-5, inclusive. The "dsc-begin" digit value must be less than or equal to the "dsc-end" digit value.

The "any-char-set" defines any single character allowed in the ’user’ token field of [RFC3261].

The "exp-char-count" defines a minimum and maximum number of times a character within the exp-char-set may be repeated, inclusive. The "exp-min" digit value must be less than or equal to the "exp-max" digit value.

6. Examples

Detailed scenario examples will be provided once the WG decides which way to go with this mechanism.

The following is an example username information document:

```
  <?xml version="1.0"?>

      <userinfo xmlns="urn:ietf:params:xml:ns:userinfo"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         version="23" state="full">

         <userlist aor="sip:ip-pbx1@ssp.example.com" state="active">

           <user id="76" state="active"
              type="e164">+12345678901</user>

           <user id="77" state="removed" type="alpha">bob</user>

           <user id="78" state="active" type="e614"
              range="[0-9]{4}">+1781555</user>

           <user id="79" state="active" type="private"
              range="[0-9]{4,10}"
              context="pbx.ssp.example.com"></user>

         </userlist>

      </userinfo>
```

7. IANA Considerations

This document makes no request of IANA yet, but will if it goes forward.

8. Security Considerations

This section is still TBD.

9. Normative References

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E.

Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3680] Rosenberg, J., "A Session Initiation Protocol (SIP) Event Package for Registrations", RFC 3680, March 2004.

[draft-gin] Roach, A. B., "Registration for Multiple Phone Numbers in the Session Initiation Protocol (SIP)", draft-ietf-martini-gin-10, October 2010.

10. Informative References

[RFC3327] Willis, D., and Hoeneisen, B., "Session Initiation Protocol (SIP) Extension Header Field for Registering Non-Adjacent Contacts", RFC 3327, December 2002.

[RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.

[RFC4244] Barnes, M. (ed.), "An Extension to the Session Initiation Protocol (SIP) for Request History Information", RFC 4244, November 2005.

[RFC5486] Malas, D., and Meyer, D., "Session Peering for Multimedia Interconnect (SPEERMINT) Terminology", RFC 5486, March 2009.

Author’s Address

Hadriel Kaplan
Acme Packet
71 Third Ave.
Burlington, MA 01803, USA
Email: hkaplan@acmepacket.com

Appendix A - Rationale for Constraining the Expansion Pattern

This document’s mechanism defines a limited set of patterns which may be used in the "<expansion>" portion of the Bulk-AoR. This is in contrast to the "Wildcarded AoR" mechanism used in some deployments, which use any regular expressions (regex) for the pattern. One of the reasons this document restricts the regex syntax is to maintain [RFC3261] compliance, which does not allow common regex characters such as '^', '[', ']','{', and '}' to appear in SIP URIs.

The other reason this document does not use any arbitrary regex is that one of the goals of this document is to be useful for an IP-PBX to determine provisioning mismatches. An arbitrary regex is typically useful for verifying a given input string matches the pattern, and not for actually determining the complete set of strings the regex pattern implies. In other words, a regex is useful for authenticating a given number matches the pattern, but not for determining what all of the provisioned numbers are.

For example, a regex syntax model for "sip:1234![5-9][0-9]*!@example.com" is useful for checking if "sip:123456@example.com" is a matching number, but is extremely difficult for an IP-PBX to verify that the SSP does not include numbers the PBX does not have provisioned. The IP-PBX could check each of its locally provisioned numbers against the regex pattern, but has no clean way to determine if the set allowed by the regex is not *greater* than its locally provisioned set.

Furthermore, numerous regex patterns can be used to mean the exact same set. For example "sip:1234!(5|6|7|8|9)[0-9]*!@example.com", "sip:1234![5-9][0-9]{0,}!@example.com", "sip:1234![5-9][[:digits:]]*!@example.com", and "sip:123!4[5-9][0-9]*!@example.com" all represent the same set of user strings as the first regex example.

Therefore, to avoid such issues, this document uses a very narrow set of possible "patterns", which can be used for both matching and provisioning verification.